Supplementary materials for: Resource Allocation and Service Provisioning Schedule for Cloud Edge based Cyber Physical Systems

P. Risha[1] and S. Gopikrishnan[2]

[1] School of Computer Science and Engineering, VIT University, Chennai-600127. Tamilnadu, India. risha.2021@vitstudent.ac.in

[2] School of Computer Science and Engineering, VIT-AP University, Amaravati-522237. Andhra Pradesh, India. gopikrishnan.s@vitap.ac.in

Abstract. The integration of Cyber-Physical Systems (CPS) with advanced computational resources and networking is critical for optimizing resource allocation and service provisioning across various sectors such as healthcare, manufacturing, and transportation. This paper explores the challenges and solutions for effective resource management in CPS, focusing on dynamic provisioning and virtualization techniques. We present a comprehensive resource provisioning architecture that includes cloud coordinators, data centers, hosts, and sensors, designed to improve resource utilization and service reliability. The proposed architecture virtualizes physical sensors into virtual sensors within a cloud environment, enabling seamless access and management. Our approach leverages algorithms for resource node selection, global resource provisioning, and load balancing, ensuring optimal resource allocation and high service availability. Through a series of case studies, we demonstrate significant improvements in resource utilization for different sensor types, highlighting the effectiveness of our techniques. This work underscores the importance of scalable, adaptive, and secure resource management strategies in enhancing the performance and reliability of CPS.

Keywords: Cyber-Physical Systems (CPS), dynamic resource provisioning, data centers, load balancing, resource allocation, service provisioning.

1 Introduction

This is a supplementary document which consists of Figures, Tables, Algorithms and Result graphs.

Algorithm 1 Resource_Provisioning (task, discoveredService)

```
Input: Task, DiscoveredService
Output: resourceNode
cpsUUID = discoveredService.cpsUUID
resourceSystemUUID = discoveredService.resourceSystemUUID
cpsSystem = GetCpsSystem (cpsUUID)
resourceSystem = GetResourceSystem (resourceSystemUUID)
resourceGeoLocation = resourceSystem.geoLocation
resourceNodeList = resourceSystem.resourceNodeList (Task)
if resourceNodeList.size > 0 then
    availableNodeList = Get AvailableResourceNodeList()
else
    return NULL
end if
for each node in availableNodeList do
    Apply Merge Sort on availableNodeList by injecting Resource_Node_Comparator (ND1, ND2)
end for
resourceNode = availableNodeList[0]
nodeAttributes = resourceNode.getAttributes()
serviceAllocationCount = nodeAttributes.getServiceAllocationCount()
serviceAllocationCount = serviceAllocationCount + 1
return resourceNode
```

Algorithm 2 Resource_Node_Comparator (ND1, ND2)

```
Input: ND1, ND2
Output: return -1 if ND1 > ND2, 1 if ND1 < ND2, 0 if ND1 = ND2
pm1 = ND1.Attributes.Model
pm2 = ND2.Attributes.Model
getAttributes of ND1
fromlocation = resourceSystem.geoLocation
distance1 = pm1.Geodistance
Geodistance1 = GeoDistance(fromLocation, distance1)
getAttributes of ND2
fromlocation = resourceSystem.geoLocation
distance2 = pm2.Geodistance
Geodistance2 = GeoDistance(fromLocation, distance2)
prepare attribute lists
Compare Node Attributes using Multi_Attribute_Comparator (ND1, ND2)
```

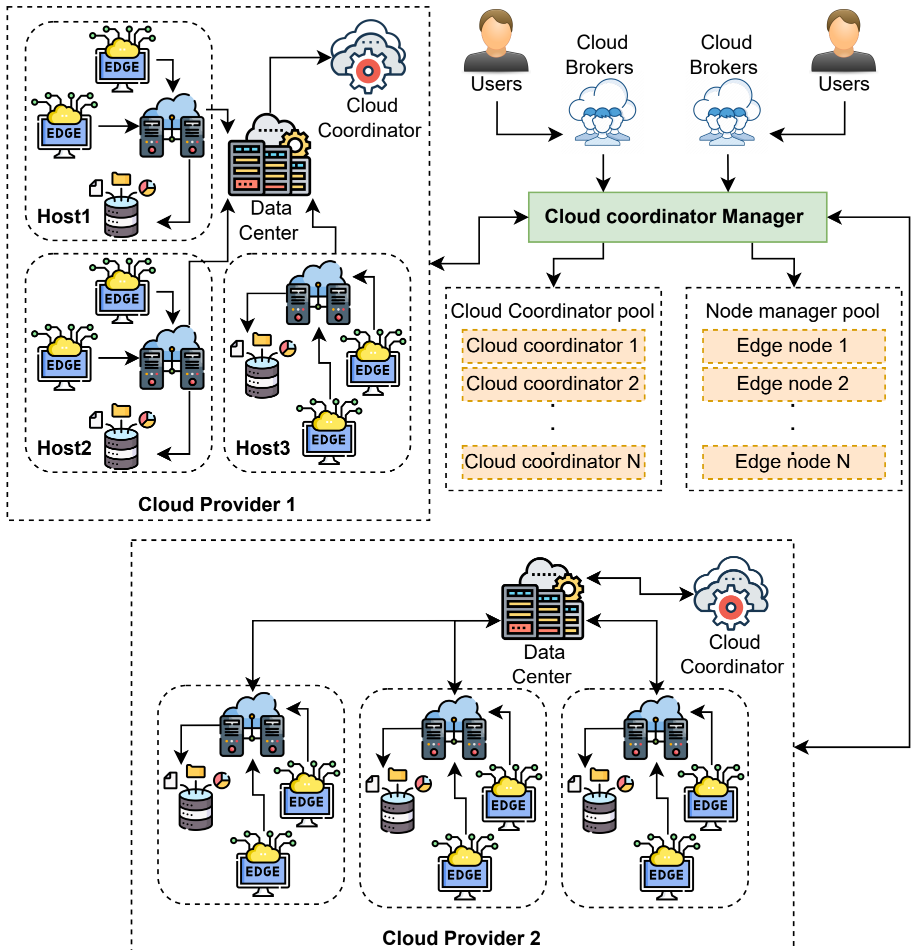

Fig. 1. Architecture of Resource Provisioner - Cloud Sensor Simulator

Table 1: Summary of Existing Resource provisional algorithm

Ref.	Method Used	Advantages	Limitations
[17]	Dynamic resource provisioning integrating cloud, fog, and edge computing	Improved system performance, resource utilization, and responsiveness	Focuses on performance improvements but neglects security considerations

Ref.	**Method Used**	**Advantages**	**Limitations**
[11]	Optimal flower pollination algorithm for resource provisioning in cyber-physical-social systems	Optimized resource allocation, improved efficiency, and reduced latency	May not scale well in very large networks
[8]	Resource offload consolidation based on deep-reinforcement learning for smart cities	Improved quality of service, reduced operational costs	Struggles with integrating strategies into existing legacy systems
[10]	Secure resource provisioning for protecting sensitive data	Enhanced data protection and security in CPS	Cybersecurity measures need more real-world testing
[3]	Robotic edge resource allocation for agricultural CPS	Efficient resource management and task allocation in agriculture	Does not fully address complexities of diverse robotic systems
[19]	Blockchain technology for enhancing security and transparency	Improved security and transparency in resource provisioning	Potential scalability issues with high transaction volumes
[4]	Energy-efficient resource management in fog computing for medical CPS	Reduced energy consumption, improved resource management	Energy-efficient model might reduce performance in some cases
[16]	Joint sampling time and resource allocation for power efficiency in industrial CPS	Optimized system performance, reduced latency	Requires more validation across different scenarios
[7]	Learning-assisted real-time resource allocation	Adapts to changing workloads, improves real-time allocation	May struggle with adaptability in highly dynamic environments
[12]	QoS-aware virtual machine scheduling for energy conservation	Ensures quality of service, optimizes resource utilization	Does not account for all possible QoS parameters
[15]	Resource provisioning in edge/fog computing for IoT-based healthcare systems	Improved efficiency and responsiveness in healthcare applications	Needs more real-world testing
[6]	Dynamic edge and cloud service integration for industrial IoT	Enhanced efficiency and responsiveness in industrial applications	Needs more real-world testing
[2]	Resource allocation in cloud-fog-edge computing for smart agriculture	Optimized resource allocation, improved efficiency in agriculture	May not effectively address all environmental variables
[1]	CyberTwin concept for resource provisioning in IoE applications	Improved resource management in 6G-enabled edge networks	Needs practical deployment for full assessment

Ref.	**Method Used**	**Advantages**	**Limitations**
[5]	Learning-based resource provisioning in fog computing	Reduced latency, improved resource allocation	Requires more real-time testing
[14]	IoT-based digital twin for energy CPS	Enhanced design and implementation of energy CPS	Needs further scalability and efficiency improvements
[9]	Ephemeral-CPS using shared devices in open IoT	Reduced resource wastage, improved efficiency	Needs further scalability improvements
[18]	Multi-objective task scheduling for fog computing	Balances performance, cost, and energy efficiency	Might not balance all objectives in complex environments effectively
[13]	GoodSpread algorithm for criticality-aware static scheduling	Improved resource allocation and system performance	Needs more benchmarking for validation

Algorithm 3 Geo_Distance (from, to)

```
Input: from, to
Output: geographical distance in meters
lat1 = from.latitude and lat2 = to.latitude
lon1 = from.longitude and lon2 = to.longitude
R = 6371000
diff = lon1 - lon2
dist = sin(lat1) * sin(lat2) + cos(lat1) * cos(lat2) * cos(diff)
dist = cos(dist)
dist = dist * R
return dist
```

References

1. Adhikari, M., Munusamy, A., Kumar, N., Srirama, S.N.: Cybertwin-driven resource provisioning for ioe applications at 6g-enabled edge networks. IEEE Transactions on Industrial Informatics **18**(7), 4850–4858 (2021)
2. Afrin, M., Jin, J., Rahman, A., Rahman, A., Wan, J., Hossain, E.: Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey. IEEE Communications Surveys & Tutorials **23**(2), 842–870 (2021)
3. Afrin, M., Jin, J., Rahman, A., Gasparri, A., Tian, Y.C., Kulkarni, A.: Robotic edge resource allocation for agricultural cyber-physical system. IEEE Transactions on Network Science and Engineering **9**(6), 3979–3990 (2021)

Algorithm 4 Global_Resource_Provisioning (Rg)

```
Input: Rg is group request of virtual node, where Rg = {Rv1, Rv2, ..., Rvi}
Output: Vng = {Gid{VNid1, VNid2, ..., VNidn}}, where Gid is UUID of virtual
node group, VNid1 is UUID of first virtual node group and so on
Cloud Coordinator Manager collects Broker List B
for each bi in B do
    Get request group Rg from bi
    Initialize Virtual node VN
    Generate group UUID and set GId = UUID
    for each Rvi in Rg do
        Vni = CALL ProvisionNode(Rvi)
        VN[i] = Vni
    end for
    Vng = {Gid{VN[1].Id, VN[2].Id, ..., VN[N].Id}}
    Set output Vng in broker bi
end for
return Vng
```

Algorithm 5 Schedule (CL)

```
Input: CL = set of cloudlet
Collect Broker List B by CCM
for each bi in B do
    get CL from bi
    for each cli in CL do
        Get Vnuuid from cli
        Vni = getVirtualNode(Vnuuid)
        cli = Vni
        dcId = Vn.dcId
        hId = Vn.hId
        Pn = getPhysicalNode(dcId, hId)
        Sc = Pn.getSchedular()
        CALL Sc.Schedule1(cli)
    end for
end for
```

Algorithm 6 Deprovision (VnUUID)

```
Input: VnUUID is UUID of virtual node
Get list of group ids (GId[ ]) from node repository
for each gid in Gid do
    VN = get virtual node list against gid
    for each Vn in VN do
        datacentre = GetDatacentre(dcId)
        host = GetHost(dcId, hId)
        node = GetNode(nId)
        removeMapping(host, node, VnUUID)
        Remove Vn from VN
        break
    end for
end for
if Vng against Gid of node repository is empty then
    remove Vng from node repository
end if
Update and save repository
```

Algorithm 7 Schedule1(cli)

```
Input: cli
Output: ND = set of records
Add cli in Q
start thread if thread is stopped or not created
while queue not empty do
    cli = get cloudlet from queue (Q)
    Vn = cli.getVirtualNode()
    dcId = Vn.dcId
    hId = Vn.hId
    nId = Vn.nId
    fromDate = cli.fromDate
    toDate = cli.toDate
    ND = load(dcId, hId, nId, fromDate, toDate)
    get broker bi from cli
    submit cloudlet data (ND) to bi
end while
stop thread
```

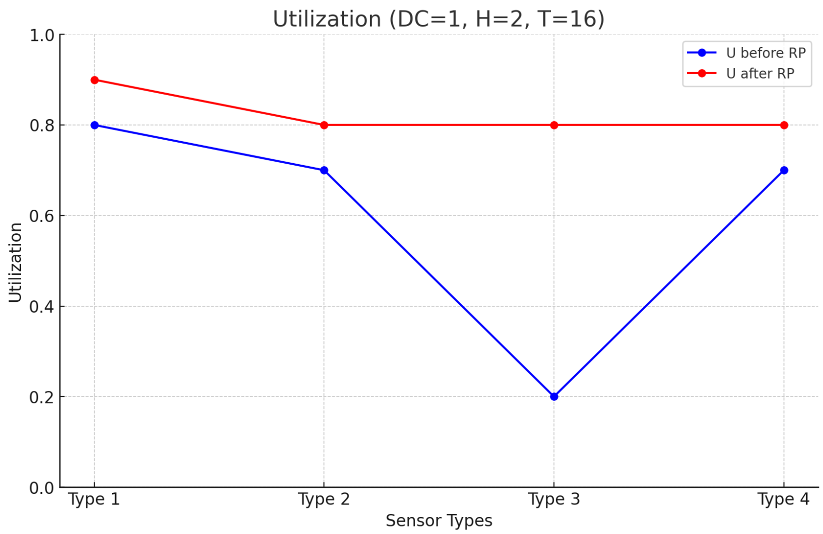

Fig. 2. Resource Utilization Case 1: DC=1, H=2, T=16

4. Apat, H.K., Bhaisare, K., Sahoo, B., Maiti, P.: Energy efficient resource management in fog computing supported medical cyber-physical system. In: 2020 International conference on computer science, Engineering and Applications (ICCSEA). pp. 1–6. IEEE (2020)
5. Etemadi, M., Ghobaei-Arani, M., Shahidinejad, A.: A learning-based resource provisioning approach in the fog computing environment. Journal of Experimental & Theoretical Artificial Intelligence **33**(6), 1033–1056 (2021)
6. Hästbacka, D., Halme, J., Barna, L., Hoikka, H., Pettinen, H., Larrañaga, M., Björkbom, M., Mesiä, H., Jaatinen, A., Elo, M.: Dynamic edge and cloud service integration for industrial iot and production monitoring applications of industrial cyber-physical systems. IEEE Transactions on Industrial Informatics **18**(1), 498–508 (2021)
7. Liu, Q., Han, T., Ansari, N.: Learning-assisted secure end-to-end network slicing for cyber-physical systems. IEEE Network **34**(3), 37–43 (2020)
8. Mekala, M.S., Jolfaei, A., Srivastava, G., Zheng, X., Anvari-Moghaddam, A., Viswanathan, P.: Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(2), 245–254 (2020)
9. Noguchi, H., Sugano, S.: Ephemeral-cyber-physical system: a cloud-like cps using shared devices in open iot. IEEE Systems Journal **14**(4), 5176–5186 (2020)
10. Okafor, K.C., Ndinechi, M.C., Misra, S.: Cyber-physical network architecture for data stream provisioning in complex ecosystems. Transactions on Emerging Telecommunications Technologies **33**(4), e4407 (2022)
11. Porkodi, V., Singh, A.R., Sait, A.R.W., Shankar, K., Yang, E., Seo, C., Joshi, G.P.: Resource provisioning for cyber–physical–social system in cloud-fog-edge computing using optimal flower pollination algorithm. ieee access **8**, 105311–105319 (2020)

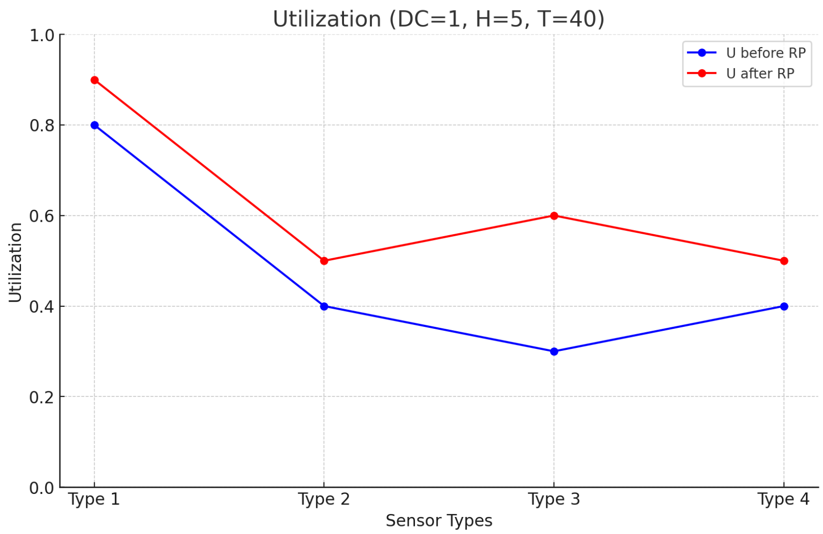

Fig. 3. Resource Utilization Case 2: DC=1, H=5, T=40

12. Qi, L., Chen, Y., Yuan, Y., Fu, S., Zhang, X., Xu, X.: A qos-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web **23**, 1275–1297 (2020)
13. Roy, D., Ghosh, S., Zhu, Q., Caccamo, M., Chakraborty, S.: Goodspread: Criticality-aware static scheduling of cps with multi-qos resources. In: 2020 IEEE Real-Time Systems Symposium (RTSS). pp. 178–190. IEEE (2020)
14. Saad, A., Faddel, S., Mohammed, O.: Iot-based digital twin for energy cyber-physical systems: design and implementation. Energies **13**(18), 4762 (2020)
15. Shakarami, A., Shakarami, H., Ghobaei-Arani, M., Nikougoftar, E., Faraji-Mehmandar, M.: Resource provisioning in edge/fog computing: A comprehensive and systematic review. Journal of Systems Architecture **122**, 102362 (2022)
16. Termehchi, A., Rasti, M.: Joint sampling time and resource allocation for power efficiency in industrial cyber–physical systems. IEEE Transactions on Industrial Informatics **17**(4), 2600–2610 (2020)
17. Xu, Z., Zhang, Y., Li, H., Yang, W., Qi, Q.: Dynamic resource provisioning for cyber-physical systems in cloud-fog-edge computing. Journal of Cloud Computing **9**, 1–16 (2020)
18. Yang, M., Ma, H., Wei, S., Zeng, Y., Chen, Y., Hu, Y.: A multi-objective task scheduling method for fog computing in cyber-physical-social services. IEEE access **8**, 65085–65095 (2020)
19. Yu, C., Jiang, X., Yu, S., Yang, C.: Blockchain-based shared manufacturing in support of cyber physical systems: concept, framework, and operation. Robotics and Computer-Integrated Manufacturing **64**, 101931 (2020)

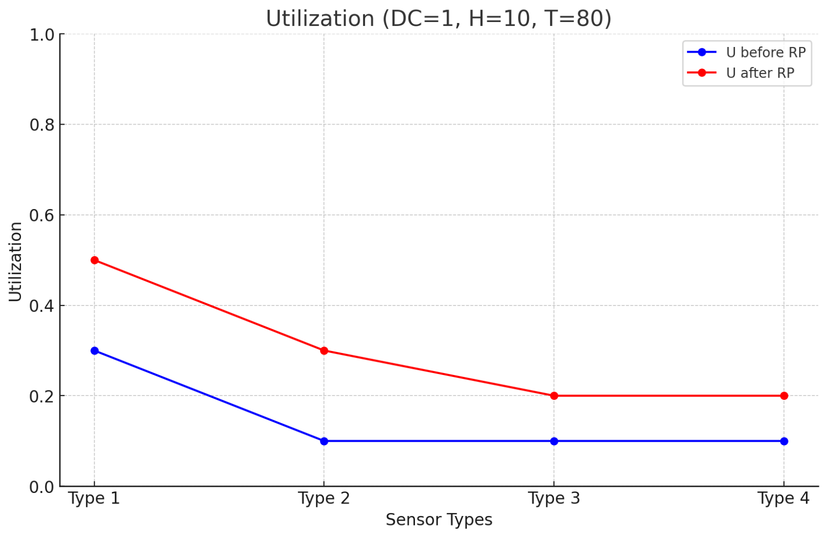

Fig. 4. Resource Utilization Case 3: DC=1, H=10, T=80

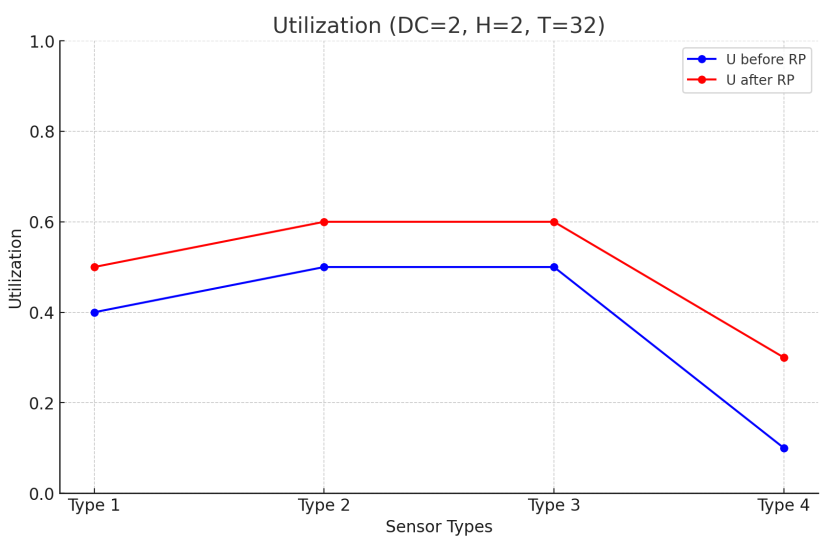

Fig. 5. Resource Utilization Case 4: DC=2, H=2, T=32

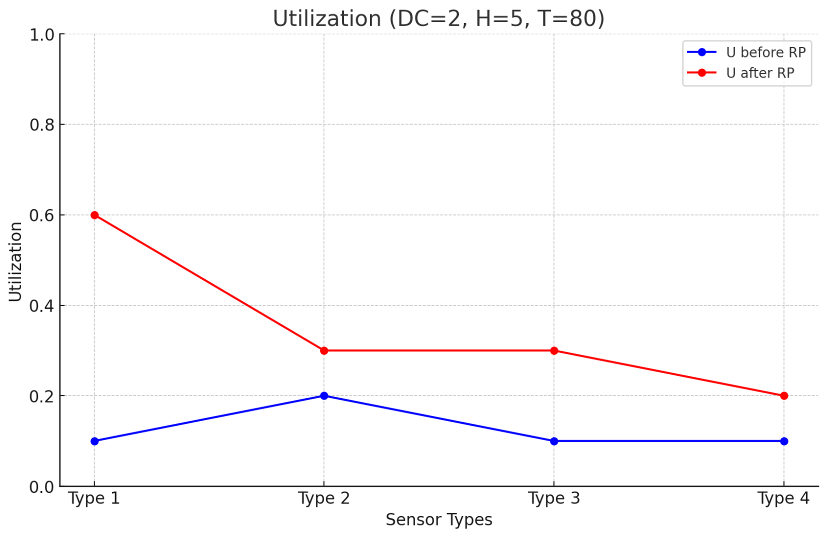

Fig. 6. Resource Utilization Case 5: DC=2, H=5, T=80

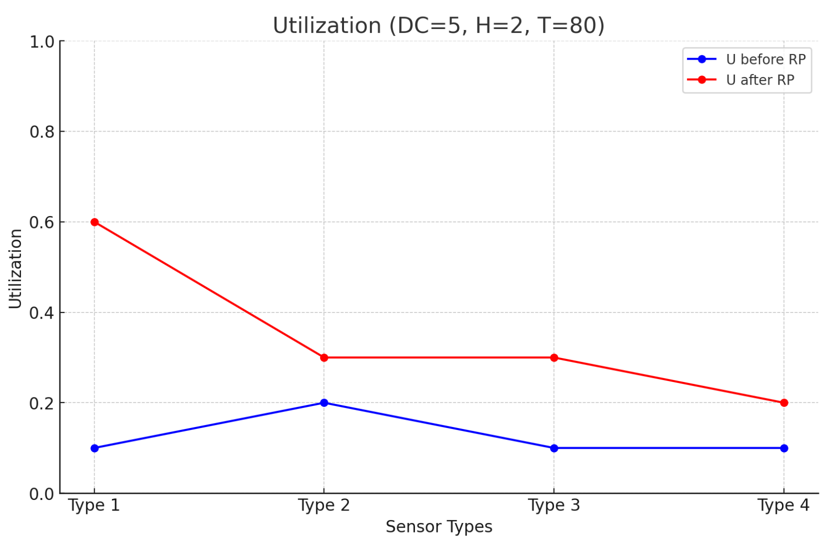

Fig. 7. Resource Utilization Case 6: DC=5, H=2, T=80

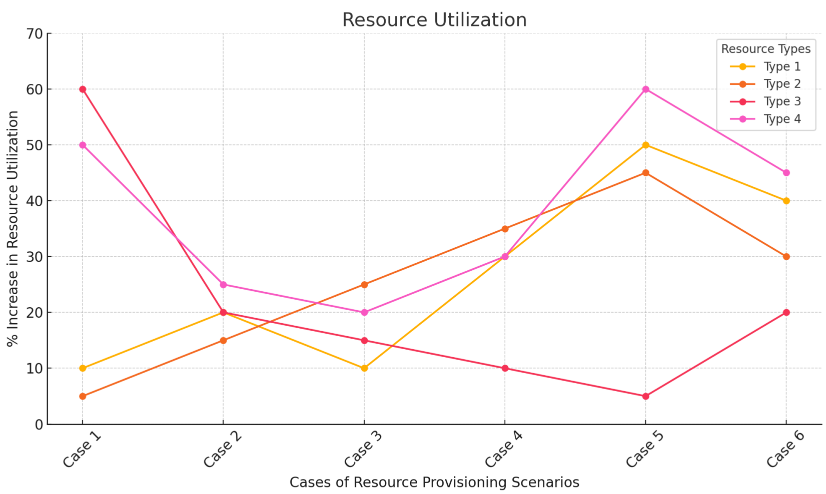

Fig. 8. Resource Utilization of All Sensor Types for Above Scenarios